

## react 실습

1>cdn 방식 이용-이클립스

```
<script src="http://react.js">  
<script src="http://react-dom.js">  
<script src="http://babel.js">  
<script type=
```

ReactDOM.render(<div>,,,,</div>);

</

2>npm 방식 이용-이클립스(node.js+react 플러그인 설치 필요)

3> npm 방식 이용-cra 툴 이용(node.js 설치 필요)

3-1. node.js 설치-자바스크립트 서버 라이브러리

3-2. 명령프롬프트

npm install -g create-react-app

3-3. 명령프롬프트

react workspace 이동

cd c:\Wkdt-venture\workspace-react

3-4. 리액트 프로젝트 생성

create-react-app **hello-react**

3-5. 리액트 프로젝트 이동

cd **hello-react**

3-6. 리액트 프로젝트 시작

npm start

브라우저

<http://localhost:3000>

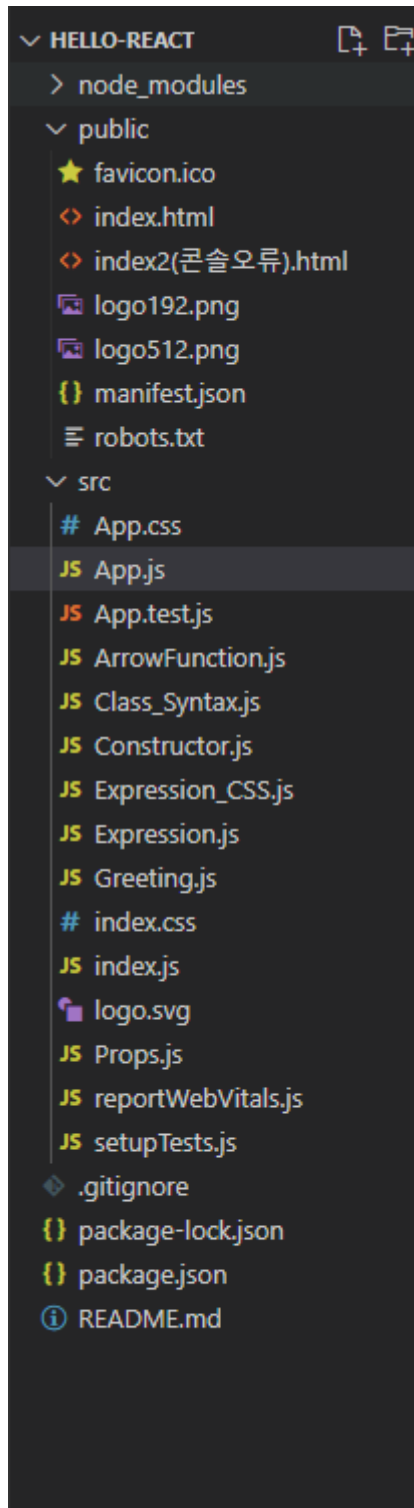
3-7. vs code

파일-폴더 열기

터미널

npm start

## react 구조



1>전체 구성 이해하고 실행하여 결과 보기

2>App.js 내부의 내용을 주석처리하고 아래와 같이 변경하기

```
function App(){  
  return (<div><h1>나의 첫번째 리액트 예제 성공</h1></div>)  
}  
export default App;
```

3> import export 이해하기

<b>App.js</b> function App(){ ... } export {App}; ->다른 js 파일로 App 이름 모듈 사용	<b>index.js</b> import {App} from './App'; ->현재폴더/App.js파일 내의 App 모듈 가져온다
다른 파일에서 사용 가능 설정	다른 파일에서 가져와서 사용

4> 다른 js 파일 정의하여 컴포넌트로 사용하기

```
function Greeting(){  
  var r = parseInt((Math.random()*3));  
  if(r==0) return <div><h1>오늘도 즐거운 하루 되세요</h1></div>;  
  if(r==1) return <div><h1>안녕하세요 리액트!!!</h1></div>;  
  if(r==2) return <div><h1>오늘 날씨는 어때요?</h1></div>;  
}  
  
function Othertest(){  
  return <div><h1>test</h1></div>;  
}  
export default Greeting;  
export {Othertest};
```

## 5> 자바스크립트 표현식 사용하기

```
const item = "SD 카드";
const value = 12000
const title ="커피 3 잔";
const imgUrl = "http://localhost:8080/images/a.png";
const msg = (<div>
    <h1>{item}-{value}</h1>
    <h1>{title}</h1>
    <p><img src={imgUrl}/></p>
</div>);
return msg;
```

## 5> css 요소 사용하기

```
function Style(){
    // 각 변수에 css 설정을 담아둡니다.
    const css1={
        'color' : '#ff0000',
        'backgroundColor' : '#f0f0ff',
        'fontSize' : '2em'
    }
    const css2={
        'color' : '#0000ff',
        'backgroundColor' : '#fff0ff',
        'fontSize' : '2em'
    }
    return(
        <div>
            <p style={css1}>스타일을 적용한 빨강색 글씨</p>
            <p style={css2}>스타일을 적용한 파랑색 글씨</p>
        </div>
    );
}

export default Style;
```

## 7> 화살표함수 선언하기

```
const Hello2=()=>{
```

```

const clickHandler=(e)=>{
  alert('화살표 함수를 이용해서 이벤트처리함!')
}
return(<div onClick={clickHandler}>
  <h1>클릭해보세요(화살표)</h1>
</div>)
}
export default Hello2;

```

## 6>class 선언하기

```

import React from 'react';
class Hello extends React.Component{
  render(){
    const clickHandler=(e)=>{
      alert(' 클래스 방식으로 이벤트처리합니다 ')
    }
    return(<div onClick={clickHandler}>
      <h1>클릭해보세요!!</h1>
    </div>)
  }
}

export default Hello;

```

## 8>props 사용하기

```

const HeaderParts = (props)=>(
  <div style={{backgroundColor:'red',color:'white'}}>
    <h3>{props.title}</h3>
  </div>
)
const BodyParts = (props)=>(
  <div style={{border:'1px solid blue',margin:15px}}>
    <h3>{props.body}</h3>
  </div>
)
const Book = (props)=>(
  <div>
    <HeaderParts title={props.title} />
    <BodyParts body={props.body} />
  </div>
)

```

```

    </div>
  )

  export default Book;

```

node-modules	접근금지
public	other.html = css html js jquery index.html <h1>,,</h1> <div id="root"> </div> <a href="http://localhost:3000">http://localhost:3000</a> <a href="http://localhost:3000/index.html">http://localhost:3000/index.html</a>
src	index.css index.js=리액트 컴포넌트 + 위치 ReactDOM.render( <A /> <B /> <C /> , document.getElementById("root") ); ); import './index.css'  App.js function A(){==> A 리액트 컴포넌트 return (<div className="">

	<pre> &lt;h1&gt;&lt;a..&gt;&lt;/a&gt;&lt;table&gt;....&lt;/div&gt;; } B.js const B = ()=&gt;( JSX 문법 내용 구성) C.js import React from 'react'; class C extends React.Component{ render(){  ...return (JSAX 문법 내용 구성)  }; } </pre>
--	--

1> export default 컴포넌트이름 ==>

2> import 컴포넌트이름 from 'js파일명(확장자제외)'

3> ReactDOM.render (컴포넌트이름, dom객체);

4> 컴포넌트 리턴값 형태 JSX 문법

5> <컴포넌트 />-닫는다

6> JSX 내부 자바스크립트 표현식, 변수 사용 { 변수명 }

7> JSX주석

```

{/*   */}

```

8> JSX 컴포넌트 속성명 두단어결합형식-카멜케이스

onClick

className

9> <JSX 컴포넌트명 style=

```
{{color:'red' , border : '2px solid blue' , backgroundColor:'green' }}
```

```
<html태그명 style="color:red;border:2px solid blue; background-color  
=green"
```

	index.js
--	----------

-

```
return (<div>
```

```
{ 10 > 5 ? `안녕하세요 ${p.name}` : `xxxx ${p.age}` }
```

```
</div>);
```

```
var a = `문자열 ${자바스크립트변수}`
```



## 9> 클래스 선언식 생성자 이용한 props 사용하기

```
class Hello3 extends React.Component{
  constructor(props){
    super(props)
  }
  render(){
    const clickHandler=(e)=>{
      const name=this.props.name + " 회원님 ";
      alert(`${name} 안녕하세요?`)
    }
    return(<div onClick={clickHandler}>
      <h1>{this.props.name}</h1>
    </div>)
  }
}

export default Hello3;
```

## 10> state 사용하기

```
class Hello3 extends React.Component{
  constructor(props){
    super(props)
    this.state = {
      value: '추가한 value 변수',
      name:this.props.name + " 회원님"
    };
  }
  render(){
    const clickHandler=(e)=>{
      const name=this.props.name + " 회원님 ";
      alert(`${name} 안녕하세요?`)
    }
    return(<div onClick={clickHandler}>
      <h1>{this.state.name }:{this.state.value}</h1>
    </div>)
  }
}
```

```
}  
export default Hello3;
```

1> ...연산자처럼 객체 원본 복사

2> 배열객체.map() 메소드처럼 배열 복사 작업

1> <BoardHeading />--> 클래스 화살표 유명함수

---> <tr><th>제목</th><th>작성자</th><th>조회수</th></tr>

2> <Board data={배열변수}/>

--> 배열 내부 게시물 데이터 ???개

--><tr><td>제목</td><th>작성자</td><td>조회수</td></tr>

3> <BoardList />

==> 1호출+ 2번호출 여러번

## Loginform.js

강사님 this.state = { id 와 password는 id 속성값인가요? name 속성값  
인가요? }

강사님, 아까전에 LoginForm.js 작성시,  
this.state = { id:"", pw: ""} 이렇게 주셨을 때 id와 pw은 index.js에서  
리액트 컴포넌트 호출? 이런 방식으로 값을 주는 것이 아니었는데, 여기  
서 변수명이 리턴 값에 지정한 id속성이나 name속성 값이랑 달라도 되  
는것입니까?

저기서 말하는 생성자 함수 안에 this.state에 있는 id와 pw은 프라퍼티  
선언은 어떤 방식으로 이어지는지 궁금합니다.

-----새로운 리액트 프로젝트 생성 , 실행-----

cd c:\Wkdt-venture\workspace\_react

create-react-app 프로젝트명

vs code 열고 폴더 열기

프로젝트명\public\index.html

프로젝트명\src\\*.js

terminal

npm start

1>

port 중복--> 다른 port 임의의 선택

또는

2> 지정 port

package.json